

Zhu
AF

PTO/SB/21 (08-00)

Approved for use through 10/31/02. OMB 0651-0031

Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Under the Paper Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

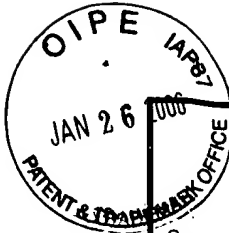
TRANSMITTAL FORM <i>(to be used for all correspondence after initial filing)</i>	Application Number	09/764,493	
	Filing Date	January 17, 2001	
	First Named Inventor	Arthur C. T. Huston	
	Group Art Unit	2155	
	Examiner Name	Bharat N. Barot	
Total Number of Pages in This Submission	42	Attorney Docket Number	50269-0039

ENCLOSURES (check all that apply)		
<input checked="" type="checkbox"/> Fee Transmittal Form <input checked="" type="checkbox"/> Fee Attached <input type="checkbox"/> Amendment / Response <input type="checkbox"/> After Final <input type="checkbox"/> Affidavits/declaration(s) <input type="checkbox"/> Extension of Time Request <input type="checkbox"/> Express Abandonment Request <input type="checkbox"/> Information Disclosure Statement <input type="checkbox"/> Certified Copy of Priority Document(s) <input type="checkbox"/> Response to Missing Parts/Incomplete Application <input type="checkbox"/> Response to Missing Parts under 37 CFR 1.52 or 1.53	<input type="checkbox"/> Assignment Papers (for an Application) <input type="checkbox"/> Drawing(s) <input type="checkbox"/> Licensing-related Papers <input type="checkbox"/> Petition <input type="checkbox"/> Petition To Convert To a Provisional Application <input type="checkbox"/> Power of Attorney, Revocation Change of Correspondence Address <input type="checkbox"/> Terminal Disclaimer <input type="checkbox"/> Request for Refund <input type="checkbox"/> CD, number of CD(s) _____	<input type="checkbox"/> After Allowance Communication to Group <input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences <input checked="" type="checkbox"/> Appeal Communication to Group (Appeal Notice, Brief, Reply Brief) <input type="checkbox"/> Proprietary Information <input type="checkbox"/> Status Letter <input checked="" type="checkbox"/> Other Enclosure(s) (please identify below): Return Postcard. _____ _____
Remarks _____ _____		

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT	
Firm or Individual name	Hickman Palermo Truong & Becker LLP Edward A. Becker, Reg. No. 37,777
Signature	
Date	January 23, 2006

CERTIFICATE OF MAILING			
I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief- Patents, Commissioner For Patents, P.O. Box 1450, Alexandria, VA 22313-1450, or facsimile transmitted to the U.S. Patent and Trademark Office on:			
Type or printed name	Angelica Maloney		
Signature		Date	January 23, 2006

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.
OID 1995-22 DIV1



Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

FEE TRANSMITTAL for FY 2006

Patent fees are subject to annual revision,
Small Entity payments must be supported by a small entity statement,
otherwise large entity fees must be paid. See Forms PTO/SB/09-12.
See 37 C.F.R. §§ 1.27 AND 1.28

Complete if Known

Application Number	09/764,493
Filing Date	January 17, 2001
First Named Inventor	Arthur C. T. Huston
Examiner Name	Bharat N. Barot
Group/Art Unit	2155
Attorney Docket No.	50269-0039

TOTAL AMOUNT OF PAYMENT (\$)**500.00**

METHOD OF PAYMENT (check one)

1. ☒ Throughout the pendency of this application, please charge any additional fees, including any required extension of time fees, and credit all overpayments to deposit account 50-1302. A duplicate of this sheet is enclosed.

Deposit Account Number

50-1302

Deposit Account Name

Hickman Palermo Truong & Becker, LLP

2. ☒ Payment Enclosed:
☒ Check ☐ Money Order ☐ Other

3. ☐ Applicant(s) is entitled to small entity status.
See 37 CFR 1.27.

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity		Small Entity		Fee Description	Fee Paid
Fee Code	Fee (\$)	Fee Code	Fee (\$)		
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet.	
1251	120	2251	60	Extension for reply within first month	
1252	450	2252	225	Extension for reply within second month	
1253	1,020	2253	510	Extension for reply within third month	
1254	1,590	2254	795	Extension for reply within fourth month	
1255	2,160	2255	1,080	Extension for reply within fifth month	
1401	500	2401	250	Notice of Appeal	
1402	500	2402	250	Filing a brief in support of an appeal	\$500.00
1452	500	2452	250	Petition to revive - unavoidable	
1453	1,500	2453	750	Petition to revive - unintentional	
1501	1,400	2501	700	Utility issue fee (or reissue)	
1502	800	2502	400	Design issue fee	
1504	300	2504	300	Publication Fee	
1462	400	1462	400	Petitions Director not specifically provided for Group I	
1463	200	1463	200	Petitions Director not specifically provided for Group II	
1464	130	1464	130	Petitions Director not specifically provided for Group III	
1806	180	1806	180	Submission of information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	790	2809	395	Filing a submission after final rejection (37 CFR § 1.129(a))	
1810	790	2810	395	For each additional invention to be examined (37 CFR § 1.129(b))	
Other fee (specify) _____					
Other fee (specify) _____					

FEE CALCULATION

1. BASIC FILING FEE

Large Entity		Small Entity		Fee Description	Fee Paid
Fee Code	Fee (\$)	Fee Code	Fee (\$)		
1011	300	2011	150	Utility filing fee	
1111	500	2111	250	Utility Search fee	
1311	200	2311	100	Utility Examination fee	
1081	250	2081	125	Utility Application Size Fee	
1005	200	2005	100	Provisional Application Fee	
1085	250	20835	125	Provisional Application Size Fee	

SUBTOTAL (1) (\$)**0.00**

2. EXTRA CLAIM FEES

	Highest Paid Claims	Extra Claims	Fee from Below	Fee Paid
Total Claims	20**	0	50.00	0.00
Independent Claims	3**	0	200.00	0.00
Multiple Dependent				

**or number previously paid, if greater; For Reissues, see below

Large Entity		Small Entity		Fee Description
Fee Code	Fee (\$)	Fee Code	Fee (\$)	
1202	50	2202	25	Claims in excess of 20
1201	200	2201	100	Independent claims in excess of 3
1203	360	2203	180	Multiple dependent claim, if not paid
1204	200	2204	100	**Reissue independent claims over original patent
1205	50	2205	25	**Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) (\$)**0.00**

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3)

(\$)**500.00**

SUBMITTED BY

Name (Print/Type)	Edward A. Becker	Registration No. (Attorney/Agent)	37,777	Telephone	(408) 414-1204
Signature		Date	January 23, 2006		

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231.
DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Mail Stop Amend, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.



**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re application of:

Confirmation No. 7203

Arthur Charles Thomas Huston, et al.

Group Art Unit No.: 2155

Serial No.: 09/764,493

Examiner: Bharat N. Barot

Filed: January 17, 2001

For: APPROACH FOR MANAGING AND PROVIDING CONTENT TO USERS

Mail Stop Appeal Brief – Patents

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

This Appeal Brief is submitted in support of the Notice of Appeal filed on November 23, 2005.

I. REAL PARTY IN INTEREST

Inktomi Corporation is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related appeals or interferences.

III. STATUS OF CLAIMS

Claims 1-90 are pending in this application. Claims 71-76 are withdrawn from consideration by virtue of the prior election for examination of Claims 1-70 and 77-90 made in response to the Requirement for Restriction that was mailed on June 28, 2004. Claims 1-70 and 77-90 were finally rejected in the Final Office Action mailed on September 29, 2005 (hereinafter “the Final Office Action”) and are the subject of this appeal.

IV. STATUS OF AMENDMENTS

No amendments were filed after the Final Office Action mailed on September 29, 2005.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present application contains independent Claims 1, 13, 25, 37, 49, 59, 69, 70, 77, 78, 79 and 90 that are the subject of this appeal. A summary of each of these claims with reference to the specification and drawings is provided hereinafter.

Independent Claim 1 is directed to a method for managing data stored in a cache that includes pre-fetching data into the cache independent of any requests for the data. According to the method recited in Claim 1, a first version of data is provided in response to receiving a first request for the data. The availability of a second more recent version of data is detected independent of any requests for the data. In response to detecting, independent of any request for the data, that the second more recent version of the data is available, a request is made to supply the second more recent version of the data to the cache. The second more recent version of the data is stored in the cache. In response to receiving a second request for the data, the second more recent version of the data is retrieved from the cache and provided. This approach for managing data stored in a cache using pre-fetching of data is described in at least Section 6 of the

Specification entitled “Cache Pre-Fetch” on Pages 21-23, and FIGS. 2A and 2B. The text on Page 13, lines 2-14 of the specification provides an example of how a differencing engine 240 can detect that a new version of data is available, independent of any requests for the data.

Independent Claim 13 recites the same limitations as Claim 1, except in the context of a computer-readable medium. Independent Claim 25 is directed to a method for managing data stored in a cache that is similar to the method of Claim 1, except in Claim 25, the first version of the data is deleted from the cache and the second version of the data is not provided in response to a second request for the data. Independent Claim 37 recites the same limitations as independent Claim 25, except in the context of a computer-readable medium. Independent Claim 49 is directed to a method for managing data stored in a cache that is similar to Claim 1, except in Claim 49, the pre-fetching is performed for one or more data items selected from a plurality of data items. Independent Claim 59 recites the same limitations as independent Claim 49, except in the context of a computer-readable medium. Independent Claim 69 is directed to a method for managing data in a cache that uses a pre-fetch approach similar to Claim 1, except in the context of pre-fetching new data that is not stored in the cache. Claim 70 recites the same limitations as independent Claim 69, except in the context of a computer-readable medium.

Independent Claim 77 recites limitations similar to Claim 1, except in the context of managing content at a traffic server that receives content from an origin server. Independent Claim 78 recites the same limitations as Claim 77, except in the context of a computer-readable medium. Independent Claim 79 is directed to an apparatus for managing content on a cache. The apparatus of Claim 79 includes a differencing engine that detects that a more recent version of data is available, where the detection is made independent of any request for the data. Independent Claim 90 is directed to an apparatus that is similar to the apparatus in Claim 79,

except that with the apparatus of Claim 90, the differencing engine is also configured to cause an older version of the data to be deleted from the cache.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1-70 and 77-90 stand finally rejected under 35 U.S.C. § 112, first paragraph, for failing to comply with the written description requirement and the enablement requirement.
2. Claims 1-70 and 77-90 stand finally rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicant regards as the invention.
3. Claims 1, 4, 5, 7, 8, 10, 13, 17, 19, 20, 22, 69 and 70 stand finally rejected under 35 U.S.C. § 102(e) as being anticipated by *Keeseey et al.*, U.S. Patent No. 6,622,167 (hereinafter “*Keeseey*”).
4. Claims 2, 3, 6, 9, 11, 12, 14, 15, 18, 21, 23-69 and 77-90 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over *Keeseey*.

VII. ARGUMENTS

- A. **Claims 1-70 and 77-90 fully comply with both the written description requirement and the enablement requirement under 35 U.S.C. § 112, first paragraph**

Claims 1-70 and 77-90 were finally rejected under 35 U.S.C. § 112, first paragraph, for failing to comply with the written description requirement and for failing to comply with the enablement requirement. The basis for the rejection is that the limitations added to the claims via claim amendments made with the Request for Continued Examination (RCE) filed on January 24, 2005 are not supported or described by the specification of the present application.

Claim 1, as amended with the filing of the RCE on January 24, 2005, is reproduced below in marked up form to show the changes that were made via amendment:

1. A method for managing data stored in a cache comprising the computer-implemented steps of:
providing a first version of data in response to receiving a first request for data;
detecting, independent of any request for the data, that a second more recent version of the data is available;
in response to ~~detecting~~ detecting, independent of any request for the data, that the second more recent version of the data is available,
requesting the second more recent version of the data be supplied to the cache, and
~~retrieving and~~ storing in the cache the second more recent version of the data;
receiving a second request for the data; and
in response to receiving the second request for the data,
retrieving the second more recent version of the data from the cache, and
providing the second more recent version of the data.

The amendments made to Claims 1 generally included two changes. The first change amended the “in response to detecting” step to include the entire antecedent basis for the prior detecting step, instead of just referring to the prior detecting step by “detecting.” More specifically, the “in response to detecting that the second more recent version of data is available” was changed to “in response to detecting, *independent of any request for the data,* that the second more recent version of data is available” (new text in italics) to more completely refer to the detecting performed in the detecting step. The text added was implicit in the claim prior to amendment, since the claim referred to the detecting step, but the claim was nevertheless amended to improve the readability and clarity of the claims. As for support in the specification, the exact claim language as amended, was provided in the specification of the present application, as originally filed, in the Summary of the Invention section on Page 7, lines 4-6.

Lines 1-10 of the Summary of the Invention section on Page 7 of the present application, as originally filed, state (lines 4-6 are italicized for convenience):

According to one aspect of the invention, a method is provided for managing data stored in a cache. According to the method, a first version of data is provided in response to receiving a first request for the data. *In response to detecting, independent of any request for the data, that a second more recent version of the data is available, the second more recent version of the data is retrieved and stored in the cache.* A second user request for the data is received. In response to receiving the second user request for the data, the second more recent version of the data is retrieved from the cache and the second more recent version of the data is provided. In another embodiment, the first version of the data is deleted from the cache.

Since the first change to Claim 1, exactly as amended, was included in the application as filed, it is respectfully submitted that the first change made to Claim 1 satisfies both the written description and enablement requirements of 35 U.S.C. § 112, first paragraph.

The second change amended the Claim 1 limitation “retrieving and storing in the cache the second more recent version of the data” to include “requesting the second more recent version of the data be supplied to the cache.” Although it is implicit that retrieving the second more recent version of the data includes requesting the second more recent version of data, this amendment was made to improve the readability and clarity of the claims. As for support in the specification, this claim language is supported by the description contained in Section 6 “Cache Pre-Fetch” starting on Page 21 of the specification. For example, the text at Page 23, lines 3-12 of the present application as filed describes how pre-fetch requests and commands are issued to retrieve new versions of data:

In addition, a pre-fetch queue may be employed to enable pre-fetch requests to be re-issued if an origin server is unavailable. For example, suppose that differencing engine 240 detects that new content is available from origin server 204. Differencing engine 240 issues a pre-fetch command to traffic server 218 to retrieve and store the new content in cache 238. Traffic server 218 issues a pre-fetch command to origin server 204, but an error or unavailability of origin server 204 prevents the new content from being provided to traffic server 218. In this situation, traffic server 218 may be configured with a pre-fetch request queue. The pre-fetch request for the new content is placed in the pre-fetch request queue for later processing, when origin server 204 becomes available.

In view of this description provided in the application as filed, it is respectfully submitted that the second change made to Claim 1 satisfies both the written description and enablement requirements of 35 U.S.C. § 112, first paragraph.

The (non-final) Office Action mailed on April 14, 2003 asserts “when reading pages 13-14 of the specification, the differencing engine always inquires about newer versions based upon the cached versions that were previously requested.” Applicant respectfully submits that the description on pages 13-14 of the specification provides examples of how the differencing engine might detect that new content is available and there is nothing in the specification that describes that the differencing engine “always inquires about newer versions based upon the cached versions that were previously requested.” To the contrary, the discussion of the pre-fetch technique in Section 6 describes how content may be pre-fetched irrespective of whether the content has been previously requested. For example, the text at Page 21, lines 13-22 states:

Cache pre-fetch may be employed for content that has been deleted from a cache, as well as for new content that may have never before been stored in cache. This approach can significantly reduce the amount of time required to provide requested content to a client since the content is already stored, i.e., pre-fetched, in the cache of a traffic server when a request for the content is received from a client. Thus, the client does not have to wait for the content to be retrieved from an origin server. For example, referring to FIG. 2, suppose that new content is created by a content provider and stored on origin server 206. According to one embodiment of the invention, differencing engine 240 detects that the new content is available on origin server 206 and requests the new content from origin server 206. The new content is stored in cache 236.

This portion of the specification clearly and unequivocally states that the cache pre-fetch approach recited in Claim 1 may be used for content that has never been stored in a cache or requested and completely refutes the Examiner’s assertion that “the differencing engine always inquires about newer versions based upon the cached versions that were previously requested.”

Independent Claims 1, 13, 25, 37, 77, 78, 79 and 90 were amended in the same manner as Claim 1 and therefore satisfy both the written description and enablement requirements of 35 U.S.C. § 112, first paragraph.

As for the other independent Claims 49, 59, 69 and 70, these claims were amended in a manner similar to independent Claims 1, 13, 25, 37, 77, 78, 79 and 90, albeit with slightly different language. It is respectfully submitted that the amendments to Claims 49, 59, 69 and 70 are also fully supported by the specification in the sections identified above with respect to Claim 1. It should be pointed out that other portions of the specification may also provide support for the claim amendments made with the filing of the RCE, but a discussion of these portions is not necessary at this time given the clear support already identified.

In view of the foregoing, it is respectfully submitted that the amendments made with the filing of the RCE are fully supported by the present application and that Claims 1-70 and 77-90 satisfy both the written description and enablement requirements of 35 U.S.C. § 112, first paragraph.

B. Claims 1-70 and 77-90 fully comply with 35 U.S.C. § 112, second paragraph

Claims 1-70 and 77-90 were rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicant regards as the invention. The stated basis for the rejection is that “[i]t is unclear how the claimed limitation of ‘in response to detecting that the second more recent version of the data is available, and requesting the second more recent version of the data be supplied to the cache’ is performed independent of any request for that data.” Page 3 of Third Office Action mailed on April 14, 2003.

It is certainly true that conventional caching mechanisms, such as those described in the *Keeseey* reference, rely upon prior requests for data to determine whether a more recent version of data is available. Unlike prior conventional approaches however, the approaches recited in Claims 1-70 and 77-90 detect that an updated version of data is available independent of any request for the data. This may be done using numerous approaches and the inventions recited in Claims 1-70 and 77-90 are not limited to any particular approaches. As one example, a notification may be received from a source of data, such as an origin server, that a new version of data has been received, independent of any request for the data. Specification at Page 13, lines 7-14. This may be used for new data that has never been previously requested or stored in a cache. Specification at Page 21, lines 11-15. This is only one example of how a second more recent version of data may be detected independent of any request for that data and the claimed invention is not limited to this example or any particular approach.

In view of the foregoing, it is respectfully submitted that Claims 1-70 and 77-90 are not indefinite for failing to particularly point out and distinctly claim the subject matter, which Applicant regards as the invention.

C. Claims 1, 4, 5, 7, 8, 10, 13, 17, 19, 20, 22, 69 and 70 are not anticipated under 35 U.S.C. § 102(e) by Keesey

i. Introduction

It is well founded that to establish a *prima facie* case of anticipation under 35 U.S.C. § 102(e), each and every element as set forth in the claim must be found, either expressly or inherently, in the single reference cited and relied upon. *Verdegaal Bros. v. Union Oil Co. of California*, 2 USPQ2d 1051 (Fed. Cir. 1987). Furthermore, “[t]he identical invention must be shown in as complete detail as is contained in the ... claim.” *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

With respect to the present application, it is respectfully submitted that *Keesey* does not teach or suggest all of the limitations of Claims 1, 4, 5, 7, 8, 10, 13, 17, 19, 20, 22, 69 and 70. It is further submitted that a sufficient factual basis has not been proffered during the prosecution of the present application to support the rejection of Claims 1, 4, 5, 7, 8, 10, 13, 17, 19, 20, 22, 69 and 70 under 35 U.S.C. § 102(e) as being anticipated by *Keesey*.

ii. Argument

SUMMARY OF KEESEY

Keesey describes an intranet server equipped with a local cache and a document shadowing program. Intranet servers so equipped are referred to as “document shadowing servers” (DSSs). The document shadowing program can be configured to operate in either a passive mode or in an active mode.

When operating in the passive mode, a DSS receives a request for a document from a downstream DSS or user. If the requested document is both a priority document and is available from the local cache of the DSS, then the DSS supplies a copy of the document to the downstream DSS or user that made the request. Otherwise, the DSS provides a request to the next upstream DSS or the Internet. When the DSS receives a response back from an upstream DSS or the Internet, the DSS determines whether the response includes a document. If so, then the DSS stores the document in its local cache if the usage count for the document equals or

exceeds a predetermined threshold level and if the document is a revised version of a document stored in the local cache. The DSS then provides a response to the downstream requestor, which may or may not include a document, depending upon whether the downstream requestor is another DSS or a user. *Keeseey*, at Col. 2, lines 26-31 and Col. 6, lines 5-47.

When operating in the active mode, a DSS is configured to periodically request new versions of documents stored in its local cache. New versions are then substituted for old versions stored in the local cache of the DSS. In the active mode, requests for new versions of documents are pushed upstream from a DSS and new versions of documents are pushed downstream from a DSS towards users. *Keeseey*, at Col. 2, lines 31-34 and Col. 7, lines 15-55.

CLAIM 1

Claim 1 is directed to a method for managing data stored in a cache that recites:
“providing a first version of data in response to receiving a first request for data;
detecting, independent of any request for the data, that a second more recent version of the data is available;
in response to detecting, independent of any request for the data, that the second more recent version of the data is available,
requesting the second more recent version of the data be supplied to the cache,
and
storing in the cache the second more recent version of the data;
receiving a second request for the data; and
in response to receiving the second request for the data,
retrieving the second more recent version of the data from the cache, and
providing the second more recent version of the data.”

It is respectfully submitted that Claim 1 is patentable over *Keeseey* because Claim 1 recites one or more limitations that are not taught or suggested by *Keeseey*. In particular, it is respectfully submitted that *Keeseey* does not teach or suggest at least the Claim 1 limitations “detecting, independent of any request for the data, that a second more recent version of the data is available” and “in response to detecting, independent of any request for the data, that the second more recent version of the data is available, requesting the second more recent version of the data be supplied to the cache, and storing in the cache the second more recent version of the data.”

In both the passive and active modes of operation of *Keeseey*, a Document Shadowing Server (DSS) determines whether a new version of a document is available only after the DSS

requests and receives a document from an upstream DSS or the Internet. After receiving a document from an upstream DSS or the Internet, the DSS then determines whether the received document is a revised version of an existing document stored in the local cache of the DSS. Thus, in *Keesey*, for a DSS to determine whether a new version of a document is available, the DSS must first make a request for the document from an upstream DSS or the Internet.

The Final Office Action refers to the text at Col. 7, lines 25-30 of *Keesey* for teaching the Claim 1 limitation “detecting, independent of any request for the data, that a second more recent version of the data is available.” Final Office Action, Page 3. The text at Col. 7, lines 25-30 of *Keesey* states “[a]lternatively, new versions could be pushed downstream to any DSS that has previously requested the document independently of the user request and inquiry process of FIGS 4 through 6.” To the extent that this portion of *Keesey* teaches or suggests the Claim 1 limitation “detecting, *independent of any request for the data*, that a second more recent version of the data is available,” it is only in the context of detecting that a more recent version of data is available independent of the user request and inquiry process of FIGS 4 through 6 of *Keesey*. This portion of *Keesey* clearly states, however, that new versions can be pushed downstream to any DSS “*that has previously requested the document.*” Thus, the detection that a new version of data is available is not made “independent of any request for the data” as recited in Claim 1, since this portion of *Keesey* describes that new versions of data are pushed to any DSS that has previously requested the document. The pushing of new data to a DSS is necessarily dependent upon the DSS having previously requested the document. There is no teaching or suggestion in *Keesey* of a DSS detecting that a newer version of a document is available, independent of any requests for the document. To summarize, when operating in the passive mode, a DSS receives a request from a downstream DSS or user and either supplies a copy of the document from its cache if one is available, or provides the request to the next upstream DSS or the Internet. Thus, when operating in the passive mode, a DSS only determines that a more recent version of a document is available in response to processing a request for the document. When operating in the active mode, a DSS periodically requests new versions of documents stored in its local cache from other DSSs or the Internet. Thus, when operating in the active mode, the DSS only determines that a more recent version of a document is available when the DSS actually makes a request for the document. In both the passive and active operating modes, the detection of a more recent version of a document is dependent upon a request for the document. It is therefore

respectfully submitted that the Claim 1 limitation “detecting, independent of any request for the data, that a second more recent version of the data is available” is not taught or suggested by *Keesey*.

Furthermore, in the situation where a DSS pushes a more recent version of a document downstream to another DSS that has previously requested the document, there is no current request that the more recent version of the document be supplied to a downstream DSS. Rather, only a prior request is made. In this situation, the Claim 1 limitation of “in response to detecting, independent of any request for the data, that the second more recent version of the data is available, *requesting the second more recent version of the data be supplied to the cache*, and storing in the cache the second more recent version of the data” is not taught or suggested by *Keesey*, since *Keesey* does not teach or suggest that a request is made to supply the more recent version of the document to a downstream DSS. An upstream DSS simply pushes a new version of a document to a downstream DSS that had previously requested the document. It is therefore respectfully submitted that the Claim 1 limitation “in response to detecting, independent of any request for the data, that the second more recent version of the data is available, *requesting the second more recent version of the data be supplied to the cache*, and storing in the cache the second more recent version of the data” is also not taught or suggested by *Keesey* (emphasis added).

In the Final Office Action, the Examiner relied upon the text at Col. 7, lines 1-30 of *Keesey* for teaching this limitation. Final Office Action, Page 3 (middle). This portion of *Keesey* describes the active mode of operation of DSSs. As described above, when operating in the active mode, a DSS periodically requests new versions of documents stored in its local cache from an upstream DSS. The upstream DSS pushes the new version of the document to the DSS that made the request. Once it receives the new version of the document, the DSS that made the request replaces the old version of the document in its cache with the new version of the document. DSSs do not request more recent versions of documents in response to detecting, independent of any requests for the documents, that new versions of the documents are available. When operating in the active mode, DSSs simply request the latest versions of documents from upstream DSSs. Their requests are not dependent upon first detecting that a more recent version of a document is available.

In the Final Office Action, the Examiner also stated that with respect to this limitation of Claim 1, “a recitation of the intended use of the claimed invention must result in a structural difference between the claimed invention and the prior art in order to patentably distinguish the claimed invention from the prior art. If the prior art structure is capable of performing the intended use, then it meets the claim.” Final Office Action, Page 3 (bottom). In response to the Examiner’s assertion, the aforementioned limitation is not an intended use of the claimed invention. Claim 1 is directed to method for managing data stored in a cache. One of the limitations of the method is that “in response to detecting, independent of any request for the data, that the second more recent version of the data is available” two functions are performed. These include “requesting the second more recent version of the data be supplied to the cache” and “storing in the cache the second more recent version of the data.” The limitation “in response to detecting, independent of any request for the data, that the second more recent version of the data is available” recites the conditions under which the two functions are performed and is not an intended use of the method. Notwithstanding the foregoing, even if Claim 1 were to recite an intended use of the claimed invention, by the Examiner’s own standard (“[i]f the prior art structure is capable of performing the intended use, then it meets the claim”), Claim 1 would still not be anticipated by *Keesey*, since, as described herein, the Intranet system of *Keesey* is not capable of performing the functionality recited in Claim 1.

Finally, in the Final Office Action, the Examiner asserted that Applicant’s arguments presented in the reply filed on July 14, 2005 “do not comply with 37 CFR 1.111 (c) because they do not clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. Further, they do not show how the amendments avoid such references or objections.” Final Office Action, Page 4. It should be initially pointed out that the reply filed on July 14, 2005 did not include any amendments, so the remark pertaining to the amendments does not apply. Furthermore, the reply filed on July 14, 2005 addressed each and every issue raised in the (non-final) Office Action mailed on April 14, 2003 and identified and discussed the same specific claim limitations that are not taught or suggested by *Keeley* as described herein.

In view of the foregoing, it is respectfully submitted that Claim 1 recites one or more limitations that are not taught or suggested by *Keesey* and that Claim 1 is therefore patentable over *Keesey*.

CLAIMS 4, 5, 7, 8 AND 10

Claims 4, 5, 7, 8 and 10 all depend from Claim 1 and include all of the limitations of Claim 1. It is therefore respectfully submitted that Claims 4, 5, 7, 8 and 10 are patentable over *Keesey* for at least the reasons set forth herein with respect to Claim 1. Furthermore, it is respectfully submitted that Claims 4, 5, 7, 8 and 10 recite additional limitations that independently render them patentable over *Keesey*.

CLAIMS 13, 17, 19, 20 AND 22

Claims 13, 17, 19, 20 and 22 recite limitations similar to Claims 1, 5, 7, 8 and 10, except in the context of computer-readable media. It is therefore respectfully submitted that Claims 13, 17, 19, 20 and 22 are patentable over *Keesey* for at least the reasons set forth herein with respect to Claims 1, 5, 7, 8 and 10.

CLAIMS 69 AND 70

Claims 69 and 70 recite limitations similar to Claim 1, except in the context of detecting whether new data that is not stored in the cache is available. It is therefore respectfully submitted that Claims 69 and 70 are patentable over *Keesey* for at least the reasons set forth herein with respect to Claim 1.

In view of the foregoing, it is respectfully submitted that Claims 1, 4, 5, 7, 8, 10, 13, 17, 19, 20, 22, 69 and 70 are patentable over *Keesey*.

D. Claims 2, 3, 6, 9, 11, 12, 14, 15, 18, 21, 23-69 and 77-90 are patentable under 35

U.S.C. § 103(a) over *Keesey*

i. Introduction

It is well founded that to establish a *prima facie* case of obviousness under 35 U.S.C. § 103(a), the references cited and relied upon must teach or suggest all the claim limitations. In addition, a sufficient factual basis to support the obviousness rejection must be proffered. *In re Freed*, 165 USPQ 570 (CCPA 1970); *In re Warner*, 154 USPQ 173 (CCPA 1967); *In re Lunsford*, 148 USPQ 721 (CCPA 1966).

With respect to the present application, it is respectfully submitted that *Keesey* does not teach or suggest all the limitations of Claims 2, 3, 6, 9, 11, 12, 14, 15, 18, 21, 23-69 and 77-90. It is further submitted that a sufficient factual basis has not been proffered during the prosecution of the present application to support the rejection of Claims 2, 3, 6, 9, 11, 12, 14, 15, 18, 21, 23-69 and 77-90 under 35 U.S.C. § 103 as being unpatentable over *Keesey*.

ii. Argument

Claims 2, 3, 6, 9, 11, 12, 14, 15, 18, 21, 23-69 and 77-90 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *Keesey*. It is respectfully submitted that Claims 2, 3, 6, 9, 11, 12, 14, 15, 18, 21, 23-69 and 77-90 are patentable over *Keesey* for at least the reasons provided hereinafter.

CLAIMS 2, 3, 6, 9, 11 AND 12

Claims 2, 3, 6, 9, 11 and 12 all depend from Claim 1 and include all of the limitations of Claim 1. It is therefore respectfully submitted that Claims 2, 3, 6, 9, 11 and 12 are patentable over *Keesey* for at least the reasons set forth herein with respect to Claim 1. Furthermore, it is respectfully submitted that Claims 2, 3, 6, 9, 11 and 12 recite additional limitations that independently render them patentable over *Keesey*.

CLAIMS 14, 15, 18, 21, 23 AND 24

Claims 14, 15, 18, 21, 23 and 24 recite limitations similar to Claims 2, 3, 6, 9, 11 and 12, except in the context of computer-readable media. It is therefore respectfully submitted that Claims 14, 15, 18, 21, 23 and 24 are patentable over *Keesey* for at least the reasons set forth herein with respect to Claims 2, 3, 6, 9, 11 and 12.

CLAIMS 25-36

Claim 25 recites “detecting, independent of any request for the data, that a second more recent version of the data is available” and “in response to detecting, independent of any request for the data, that the second more recent version of the data is available, ... and requesting the second more recent version of the data be supplied to the cache.” These limitations are recited in Claim 1 and, as set forth herein with respect to Claim 1, it is respectfully submitted that these

limitations are not taught or suggested by *Keesey*. It is therefore respectfully submitted that Claim 25 is patentable over *Keesey* for at least the reasons set forth herein with respect to Claim 1. Claims 26-36 depend from Claim 25 and include all of the limitations of Claim 25. It is therefore respectfully submitted that Claims 26-36 are also patentable over *Keesey*.

CLAIMS 37-48

Claims 37-48 recite limitations similar to Claims 25-36, except in the context of computer-readable media. It is therefore respectfully submitted that Claims 37-48 are patentable over *Keesey* for at least the reasons set forth herein with respect to Claims 25-36.

CLAIMS 49-58

Claim 49 recites limitations similar to Claim 1. For example, Claim 49 recites “determining, for each of the one or more data items, independent of any request for any of the one or more data items, whether a newer version of the data item is available” and “for each of the one or more data items where a determination is made, independent of any request for any of the one or more data items, that a newer version of the data item is available, ... requesting the newer version of the data item be supplied to the cache.” It is therefore respectfully submitted that Claim 49 is patentable over *Keesey* for at least the reasons set forth herein with respect to Claim 1. Claims 50-58 all depend from Claim 49 and include all of the limitations of Claim 49. It is therefore respectfully submitted that Claims 50-58 are also patentable over *Keesey*.

CLAIMS 59-68

Claims 59-68 recite limitations similar to Claims 49-58, except in the context of computer-readable media. It is therefore respectfully submitted that Claims 59-68 are patentable over *Keesey* for at least the reasons set forth herein with respect to Claims 49-58.

CLAIMS 69 AND 70

Claim 69 recite limitations similar to Claim 1. For example, Claim 69 recites “detecting, independent of any request for data, that new data that is not stored in the cache is available” and “in response to detecting, independent of any request for data, that the new data is available, requesting that the new data be supplied to the cache.” It is therefore respectfully submitted that Claim 69 is patentable over *Keesey* for at least the reasons set forth herein with respect to Claim

1. Claim 70 recites limitations similar to Claim 69, except in the context of a computer-readable medium. It is therefore respectfully submitted that Claim 70 is also patentable over *Keesey*.

CLAIMS 77 AND 78

Claim 77 recites limitations similar to Claim 1. For example, Claim 77 recites “detecting, independent of any request for the content, that a second more recent version of the content is available on the origin server” and “in response to detecting, independent of any request for the content, that the second more recent version of the content is available on the origin server,... requesting and receiving the second more recent version of the content from the origin server.” It is therefore respectfully submitted that Claim 77 is patentable over *Keesey* for at least the reasons set forth herein with respect to Claim 1. Claim 78 recites limitations similar to Claim 77, except in the context of a computer-readable medium. It is therefore respectfully submitted that Claim 78 is also patentable over *Keesey*.

CLAIMS 79-89

Claim 79 recites limitations similar to Claim 1. For example, Claim 79 recites “detect, independent of any request for content, that a second more recent version of the content is available” and “in response to detecting, independent of any request for the content, that the second more recent version of the content is available on the origin server, request the second more recent version of the content be supplied to the cache.” It is therefore respectfully submitted that Claim 79 is patentable over *Keesey* for at least the reasons set forth herein with respect to Claim 1. Claims 80-89 all depend from Claim 79 and include all of the limitations of Claim 79. It is therefore respectfully submitted that Claims 80-89 are also patentable over *Keesey*.

CLAIM 90

Claim 90 recites limitations similar to Claim 1. For example, Claim 90 recites “detect, independent of any requests for data stored in the cache, that a second more recent version of the data is available” and “in response to detecting, independent of any requests for data stored in the cache, that the second more recent version of the data is available, ..., and requesting the second more recent version of the data be supplied to the cache.” It is therefore respectfully submitted

that Claim 90 is patentable over *Keesey* for at least the reasons set forth herein with respect to Claim 1.

In view of the foregoing, it is respectfully submitted that Claims 2, 3, 6, 9, 11, 12, 14, 15, 18, 21, 23-69 and 77-90 are patentable over *Keesey* since each of these claims include one or more limitations that are not in any way taught or suggested by *Keesey*.

VIII. CONCLUSION AND PRAYER FOR RELIEF

Based on the foregoing, it is respectfully submitted that the rejection of Claims Claims 1, 4, 5, 7, 8, 10, 13, 17, 19, 20, 22, 69 and 70 under 35 U.S.C. § 102(e) being anticipated by *Keesey* lacks the requisite factual and legal bases. Appellants therefore respectfully request that the Honorable Board reverse the rejection of Claims 1, 4, 5, 7, 8, 10, 13, 17, 19, 20, 22, 69 and 70 under 35 U.S.C. § 102(e) by *Keesey*. It is further respectfully submitted that the rejection of Claims 2, 3, 6, 9, 11, 12, 14, 15, 18, 21, 23-69 and 77-90 under 35 U.S.C. § 103(a) as being unpatentable over *Keesey*, lacks the requisite factual and legal bases. Appellants therefore respectfully request that the Honorable Board reverse the rejection of Claims 2, 3, 6, 9, 11, 12, 14, 15, 18, 21, 23-69 and 77-90 under 35 U.S.C. § 103 over *Keesey*.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP



Edward A. Becker

Reg. No. 37,777

Date: January 23, 2006

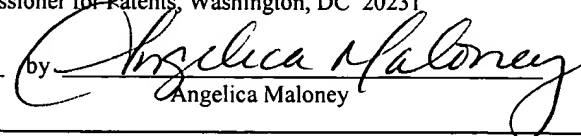
2055 Gateway Place, Suite 550
San Jose, California 95110-1089
Tel: (408) 414-1204
Fax: (408) 414-1076

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: **Mail Stop Appeal Brief – Patents**, Commissioner for Patents, Washington, DC 20231

on January 23, 2006

by


Angelica Maloney

CLAIMS APPENDIX

1. A method for managing data stored in a cache comprising the computer-implemented steps of:
providing a first version of data in response to receiving a first request for data;
detecting, independent of any request for the data, that a second more recent version of the data is available;
in response to detecting, independent of any request for the data, that the second more recent version of the data is available,
requesting the second more recent version of the data be supplied to the cache,
and
storing in the cache the second more recent version of the data;
receiving a second request for the data; and
in response to receiving the second request for the data,
retrieving the second more recent version of the data from the cache, and
providing the second more recent version of the data.
2. The method as recited in Claim 1, further comprising the step of deleting the first version of the data from the cache.
3. The method as recited in Claim 2, further comprising the steps of:
storing, in a location other than the first cache, a request to delete the first version of the data from the cache, and
if the request to delete the first version of the data from the cache cannot be successfully processed, then after expiration of a specified period of time,
retrieving from the location other than the first cache, the request to delete the first version of the data from the cache, and
processing again the request to delete the first version of the data from the cache.

4. The method as recited in Claim 1, further comprising the step of if the second more recent version of the data cannot be retrieved and stored in the cache, then after a specified period of time, attempting to again retrieve and store in the cache the second more recent version of the data.
5. The method as recited in Claim 1, further comprising the step of if, after expiration of a specified period of time from a time when the second more recent version of the data is stored in the cache, no further requests for the second more recent version of the data are received, then deleting the second more recent version of the data from the cache.
6. The method as recited in Claim 1, further comprising the steps of:
storing, in a location other than the first cache, a request to retrieve and store in the cache the second more recent version of the data, and
if the request to retrieve and store in the cache the second more recent version of the data cannot be processed successfully, then after expiration of a specified period of time,
retrieving from the location other than the first cache, the request to
retrieve and store in the cache the second more recent version of
the data, and
processing the request to retrieve and store in the cache the second more
recent version of the data.
7. The method as recited in Claim 1, further comprising providing data that indicates whether the second more recent version of the data was successfully retrieved and stored in the cache.
8. The method as recited in Claim 1, further comprising causing a copy of the second more recent version of the data to be stored at a second cache.

9. The method as recited in Claim 1, wherein the step retrieving and storing in the cache the second more recent version of the data is performed in response to processing one or more requests from an authorized entity.
10. The method as recited in Claim 1, further comprising generating, based upon a set of logging criteria, log data that indicates one or more activities of the cache.
11. The method as recited in Claim 10, wherein the set of logging criteria includes the size of the first version of data provided.
12. The method as recited in Claim 10, wherein the set of logging criteria includes an amount of time required to provide the first version of data.
13. A computer-readable medium carrying one or more sequences of one or more instructions for managing data stored in a cache, wherein execution of the one or more sequences of one or more instructions by one or more processors cause the one or more processors to perform the steps of:
 - providing a first version of data in response to receiving a first request for data;
 - detecting, independent of any request for the data, that a second more recent version of the data is available;
 - in response to detecting, independent of any request for the data, that the second more recent version of the data is available,
 - requesting the second more recent version of the data be supplied to the cache,
 - and
 - storing in the cache the second more recent version of the data;
 - receiving a second request for the data; and
 - in response to receiving the second request for the data,
 - retrieving the second more recent version of the data from the cache, and
 - providing the second more recent version of the data.

14. The computer-readable medium as recited in Claim 13, further comprising one or more additional instructions which, when executed by the one or more processors, cause the one or more processors to perform the step of deleting the first version of the data from the cache.
15. The computer-readable medium as recited in Claim 14, further comprising one or more additional instructions which, when executed by the one or more processors, cause the one or more processors to perform the steps of:
storing, in a location other than the first cache, a request to delete the first version of the data from the cache, and
if the request to delete the first version of the data from the cache cannot be successfully processed, then after expiration of a specified period of time,
retrieving from the location other than the first cache, the request to delete the first version of the data from the cache, and
processing again the request to delete the first version of the data from the cache.
16. The computer-readable medium as recited in Claim 13, further comprising one or more additional instructions which, when executed by the one or more processors, cause the one or more processors to perform the step of if the second more recent version of the data cannot be retrieved and stored in the cache, then after a specified period of time, attempting to again retrieve and store in the cache the second more recent version of the data.
17. The computer-readable medium as recited in Claim 13, further comprising one or more additional instructions which, when executed by the one or more processors, cause the one or more processors to perform the step of if, after expiration of a specified period of time from a time when the second more recent version of the data is stored in the cache, no further requests for the second more recent version of the data are received, then deleting the second more recent version of the data from the cache.

18. The computer-readable medium as recited in Claim 13, further comprising one or more additional instructions which, when executed by the one or more processors, cause the one or more processors to perform the steps of:
storing, in a location other than the first cache, a request to retrieve and store in the cache the second more recent version of the data, and
if the request to retrieve and store in the cache the second more recent version of the data cannot be processed successfully, then after expiration of a specified period of time,
retrieving from the location other than the first cache, the request to
retrieve and store in the cache the second more recent version of
the data, and
processing the request to retrieve and store in the cache the second more recent version of the data.
19. The computer-readable medium as recited in Claim 13, further comprising one or more additional instructions which, when executed by the one or more processors, cause the one or more processors to perform the step of providing data that indicates whether the second more recent version of the data was successfully retrieved and stored in the cache.
20. The computer-readable medium as recited in Claim 13, further comprising one or more additional instructions which, when executed by the one or more processors, cause the one or more processors to perform the step of causing a copy of the second more recent version of the data to be stored at a second cache.
21. The computer-readable medium as recited in Claim 13, wherein the step retrieving and storing in the cache the second more recent version of the data is performed in response to processing one or more requests from an authorized entity.
22. The computer-readable medium as recited in Claim 13, further comprising one or more additional instructions which, when executed by the one or more processors, cause the

one or more processors to perform the step of generating, based upon a set of logging criteria, log data that indicates one or more activities of the cache.

23. The computer-readable medium as recited in Claim 22, wherein the set of logging criteria includes the size of the first version of data provided.
24. The computer-readable medium as recited in Claim 22, wherein the set of logging criteria includes an amount of time required to provide the first version of data.
25. A method for managing data stored in a cache comprising the computer-implemented steps of:
providing, from a cache, a first version of data in response to receiving a first request for data;
detecting, independent of any request for the data, that a second more recent version of the data is available;
in response to detecting, independent of any request for the data, that the second more recent version of the data is available,
deleting the first version of the data from the cache, and
requesting the second more recent version of the data be supplied to the cache.
26. The method as recited in Claim 25, further comprising:
retrieving and storing in the cache the second more recent version of the data;
receiving a second request for the data; and
in response to receiving the second request for the data,
retrieving the second more recent version of the data from the cache, and
providing the second more recent version of the data.
27. The method as recited in Claim 25, further comprising the steps of:
storing, in a location other than the first cache, a request to delete the first version of the data from the cache, and

if the request to delete the first version of the data from the cache cannot be successfully processed, then after expiration of a specified period of time,
retrieving from the location other than the first cache, the request to delete
the first version of the data from the cache, and
processing again the request to delete the first version of the data from the
cache.

28. The method as recited in Claim 26, further comprising the step of if the second more recent version of the data cannot be retrieved and stored in the cache, then after a specified period of time, attempting to again retrieve and store in the cache the second more recent version of the data.
29. The method as recited in Claim 26, further comprising the step of if, after expiration of a specified period of time from a time when the second more recent version of the data is stored in the cache, no further requests for the second more recent version of the data are received, then deleting the second more recent version of the data from the cache.
30. The method as recited in Claim 26, further comprising the steps of:
storing, in a location other than the first cache, a request to retrieve and store in the cache
the second more recent version of the data, and
if the request to retrieve and store in the cache the second more recent version of the data
cannot be processed successfully, then after expiration of a specified period of
time,
retrieving from the location other than the first cache, the request to
retrieve and store in the cache the second more recent version of
the data, and
processing the request to retrieve and store in the cache the second more
recent version of the data.
31. The method as recited in Claim 25, further comprising providing data that indicates
whether the first recent version of the data was successfully deleted from the cache.

32. The method as recited in Claim 26, further comprising causing a copy of the second more recent version of the data to be stored at a second cache.
33. The method as recited in Claim 26, wherein the step retrieving and storing in the cache the second more recent version of the data is performed in response to processing one or more requests from an authorized entity.
34. The method as recited in Claim 25, further comprising generating, based upon a set of logging criteria, log data that indicates one or more activities of the cache.
35. The method as recited in Claim 34, wherein the set of logging criteria includes the size of the first version of data provided.
36. The method as recited in Claim 34, wherein the set of logging criteria includes an amount of time required to provide the first version of data.
37. A computer-readable medium carrying one or more sequences of instructions for managing data stored in a cache, wherein execution of the one or more sequences of one or more instructions by one or more processors cause the one or more processors to perform the steps of:
 - providing, from a cache, a first version of data in response to receiving a first request for data;
 - detecting, independent of any request for the data, that a second more recent version of the data is available;
 - in response to detecting, independent of any request for the data, that the second more recent version of the data is available,
 - deleting the first version of the data from the cache, and
 - requesting the second more recent version of the data be supplied to the cache.

38. The computer-readable medium as recited in Claim 37, further comprising one or more additional instructions which, when executed by the one or more processors, cause the one or more processors to perform the steps of:
retrieving and storing in the cache the second more recent version of the data;
receiving a second request for the data; and
in response to receiving the second request for the data,
retrieving the second more recent version of the data from the cache, and
providing the second more recent version of the data.
39. The computer-readable medium as recited in Claim 37, further comprising one or more additional instructions which, when executed by the one or more processors, cause the one or more processors to perform the steps of:
storing, in a location other than the first cache, a request to delete the first version of the data from the cache, and
if the request to delete the first version of the data from the cache cannot be successfully processed, then after expiration of a specified period of time,
retrieving from the location other than the first cache, the request to delete the first version of the data from the cache, and
processing again the request to delete the first version of the data from the cache.
40. The computer-readable medium as recited in Claim 38, further comprising one or more additional instructions which, when executed by the one or more processors, cause the one or more processors to perform the step of if the second more recent version of the data cannot be retrieved and stored in the cache, then after a specified period of time, attempting to again retrieve and store in the cache the second more recent version of the data.
41. The computer-readable medium as recited in Claim 38, further comprising one or more additional instructions which, when executed by the one or more processors, cause the one or more processors to perform the step of if, after expiration of a specified period of

time from a time when the second more recent version of the data is stored in the cache, no further requests for the second more recent version of the data are received, then deleting the second more recent version of the data from the cache.

42. The computer-readable medium as recited in Claim 38, further comprising one or more additional instructions which, when executed by the one or more processors, cause the one or more processors to perform the steps of:
storing, in a location other than the first cache, a request to retrieve and store in the cache the second more recent version of the data, and
if the request to retrieve and store in the cache the second more recent version of the data cannot be processed successfully, then after expiration of a specified period of time,
retrieving from the location other than the first cache, the request to
retrieve and store in the cache the second more recent version of
the data, and
processing the request to retrieve and store in the cache the second more recent version of the data.
43. The computer-readable medium as recited in Claim 37, further comprising one or more additional instructions which, when executed by the one or more processors, cause the one or more processors to perform the step of providing data that indicates whether the first recent version of the data was successfully deleted from the cache.
44. The computer-readable medium as recited in Claim 38, further comprising one or more additional instructions which, when executed by the one or more processors, cause the one or more processors to perform the step of causing a copy of the second more recent version of the data to be stored at a second cache.
45. The computer-readable medium as recited in Claim 38, wherein the step retrieving and storing in the cache the second more recent version of the data is performed in response to processing one or more requests from an authorized entity.

46. The computer-readable medium as recited in Claim 37, further comprising one or more additional instructions which, when executed by the one or more processors, cause the one or more processors to perform the step of generating, based upon a set of logging criteria, log data that indicates one or more activities of the cache.
47. The computer-readable medium as recited in Claim 46, wherein the set of logging criteria includes the size of the first version of data provided.
48. The computer-readable medium as recited in Claim 46, wherein the set of logging criteria includes an amount of time required to provide the first version of data.
49. A method for managing data stored in a cache comprising the computer-implemented steps of:
selecting, based upon one or more selection criteria, one or more data items from a plurality of data items stored on the cache;
determining, for each of the one or more data items, independent of any request for any of the one or more data items, whether a newer version of the data item is available;
and
for each of the one or more data items where a determination is made, independent of any request for any of the one or more data items, that a newer version of the data item is available,
deleting the data item from the cache,
requesting the newer version of the data item be supplied to the cache, and
storing in the cache the newer version of the data item.
50. The method as recited in Claim 49, wherein the one or more selection criteria include a source of each of the plurality of data items.
51. The method as recited in Claim 49, wherein the one or more selection criteria include a type of each of the plurality of data items.

52. The method as recited in Claim 49, wherein the one or more selection criteria include one or more users of the plurality of data items.
53. The method as recited in Claim 49, wherein the one or more selection criteria include a size of each of the plurality of data items.
54. The method as recited in Claim 49, wherein the one or more selection criteria include an age of each of the plurality of data items.
55. The method as recited in Claim 49, further comprising:
storing a request to delete the data item from the cache,
if the request to delete the data from the cache is not successfully processed, then
processing the stored request to delete the data from the cache.
56. The method as recited in Claim 49, further comprising:
storing a request to retrieve and store in the cache the newer version of the data item,
if the request to retrieve and store in the cache the newer version of the data item is not
successfully processed, then processing the stored request to retrieve and store in
the cache the newer version of the data item.
57. The method as recited in Claim 49, further comprising for each of the one or more data items where a newer version of the data item is available,
storing in a second cache the newer version of the data item.
58. The method as recited in Claim 49, wherein the steps of deleting the data item from the cache, and retrieving and storing in the cache the newer version of the data item are preformed in response to processing one or more requests from an authorized entity.

59. A computer-readable medium carrying one or more sequences of one or more instructions for managing data stored in a cache, wherein execution of the one or more sequences of one or more instructions cause one or more processors to perform the steps of:
selecting, based upon one or more selection criteria, one or more data items from a plurality of data items stored on the cache;
determining, for each of the one or more data items, independent of any request for any of the one or more data items, whether a newer version of the data item is available;
and
for each of the one or more data items where a determination is made, independent of any request for any of the one or more data items, that a newer version of the data item is available,
deleting the data item from the cache,
requesting the newer version of the data item be supplied to the cache, and
storing in the cache the newer version of the data item.
60. The computer-readable medium as recited in Claim 59, wherein the one or more selection criteria include a source of each of the plurality of data items.
61. The computer-readable medium as recited in Claim 59, wherein the one or more selection criteria include a type of each of the plurality of data items.
62. The computer-readable medium as recited in Claim 59, wherein the one or more selection criteria include one or more users of the plurality of data items.
63. The computer-readable medium as recited in Claim 59, wherein the one or more selection criteria include a size of each of the plurality of data items.
64. The computer-readable medium as recited in Claim 59, wherein the one or more selection criteria include an age of each of the plurality of data items.

65. The computer-readable medium as recited in Claim 59, further comprising one or more additional instructions which, when executed by the one or more processors, cause the one or more processors to perform the steps of:
storing a request to delete the data item from the cache,
if the request to delete the data from the cache is not successfully processed, then
processing the stored request to delete the data from the cache.
66. The computer-readable medium as recited in Claim 59, further comprising one or more additional instructions which, when executed by the one or more processors, cause the one or more processors to perform the steps of:
storing a request to retrieve and store in the cache the newer version of the data item,
if the request to retrieve and store in the cache the newer version of the data item is not
successfully processed, then processing the stored request to retrieve and store in
the cache the newer version of the data item.
67. The computer-readable medium as recited in Claim 59, further comprising one or more additional instructions which, when executed by the one or more processors, cause the one or more processors to perform the step of for each of the one or more data items
where a newer version of the data item is available,
storing in a second cache the newer version of the data item.
68. The computer-readable medium as recited in Claim 59, wherein the steps of deleting the data item from the cache, and retrieving and storing in the cache the newer version of the data item are preformed in response to processing one or more requests from an authorized entity.
69. A method for managing a cache comprising the computer-implemented steps of:
detecting, independent of any request for data, that new data that is not stored in the cache
is available;
in response to detecting, independent of any request for data, that the new data is
available,

requesting that the new data be supplied to the cache, and
storing the new data in the cache;
receiving from a user a request for the new data; and
in response to receiving the request for the new data,
retrieving the new data from the cache, and
providing the new data to the user.

70. A computer-readable medium carrying one or more sequences of one or more instructions for managing a cache, wherein execution of the one or more sequences of one or more instructions by one or more processors cause the one or more processors to perform the steps of:
detecting, independent of any request for data, that new data that is not stored in the cache is available;
in response to detecting, independent of any request for data, that the new data is available,
requesting that the new data be supplied to the cache, and
storing the new data in the cache;
receiving from a user a request for the new data; and
in response to receiving the request for the new data,
retrieving the new data from the cache, and
providing the new data to the user.

77. A method for managing content comprising the computer-implemented steps of:
retrieving from an origin server a first version of content;
storing the first version of the content on a storage medium at a traffic server;
in response to a first request for the content, retrieving the first version of the content from the storage medium and providing the first version of the content;
detecting, independent of any request for the content, that a second more recent version of the content is available on the origin server;
in response to detecting, independent of any request for the content, that the second more recent version of the content is available on the origin server,

deleting the first version of the content from the storage medium,
requesting and receiving the second more recent version of the content from the
origin server, and
storing the second more recent version of the content on the storage medium;
in response to a second request for the content, retrieving the second more recent version
of the content from the storage medium and providing the second more recent
version of the content.

78. A computer-readable medium carrying one or more sequences of one or more instructions for managing content, wherein execution of the one or more sequences of one or more instructions by one or more processors cause the one or more processors to perform the steps of:

retrieving from an origin server a first version of content;
storing the first version of the content on a storage medium at a traffic server;
in response to a first request for the content, retrieving the first version of the content
from the storage medium and providing the first version of the content;
detecting, independent of any request for the content, that a second more recent version of
the content is available on the origin server;
in response to detecting, independent of any request for the content, that the second more
recent version of the content is available on the origin server,
deleting the first version of the content from the storage medium,
requesting and receiving the second more recent version of the content from the
origin server, and
storing the second more recent version of the content on the storage medium;
in response to a second request for the content, retrieving the second more recent version
of the content from the storage medium and providing the second more recent
version of the content.

79. An apparatus for managing content on a cache comprising:
a communications interface configured to communicate with the cache; and

a differencing mechanism communicatively coupled to the communications interface and configured to
detect, independent of any request for content, that a second more recent version of the content is available,
in response to detecting, independent of any request for content, that the second more recent version of the content is available,
request the second more recent version of the content be supplied to the cache, and
cause the second more recent version of the content to be stored on the cache.

80. The apparatus as recited in Claim 79, wherein the differencing mechanism is further configured to delete from the cache a first older version of the content.

81. The apparatus as recited in Claim 80, wherein the differencing mechanism is further configured to:
store a request to delete the first older version of the content from the cache, and
if the request to delete the first older version of the content from the cache cannot be successfully processed, then after expiration of a specified period of time,
retrieving the request to delete the first older version of the content from the cache, and
processing the request to delete the first older version of the content from the cache.

82. The apparatus as recited in Claim 79, wherein the differencing mechanism is further configured to if the second more recent version of the content cannot be retrieved and stored in the cache, then after a specified period of time, attempt to again retrieve and store in the cache the second more recent version of the content.

83. The apparatus as recited in Claim 79, wherein the differencing mechanism is further configured to if, after expiration of a specified period of time from a time when the

second more recent version of the content is stored in the cache, no further requests for the second more recent version of the content are received, then deleting the second more recent version of the content from the cache.

84. The apparatus as recited in Claim 79, wherein the differencing mechanism is further configured to:

store a request to retrieve and store in the cache the second more recent version of the content, and

if the request to retrieve and store in the cache the second more recent version of the content cannot be processed successfully, then after expiration of a specified period of time,

retrieving the request to retrieve and store in the cache the second more recent version of the content, and

processing the request to retrieve and store in the cache the second more recent version of the content.

85. The apparatus as recited in Claim 79, wherein the differencing mechanism is further configured to generate data that indicates whether the second more recent version of the content was successfully retrieved and stored in the cache.

86. The apparatus as recited in Claim 79, wherein the differencing mechanism is further configured to cause a copy of the second more recent version of the content to be stored at a second cache.

87. The apparatus as recited in Claim 79, wherein the retrieving and storing in the cache the second more recent version of the content is performed in response to processing one or more requests from an authorized entity.

88. The apparatus as recited in Claim 79, wherein the differencing mechanism is further configured to generate, based upon a set of logging criteria, log data that indicates one or more activities of the cache.

89. The apparatus as recited in Claim 84, wherein the set of logging criteria includes the size of the first version of content provided.
90. An apparatus for managing a cache comprising:
a communications interface configured to communicate with the cache; and
a differencing mechanism communicatively coupled to the communications interface and configured to
detect, independent of any requests for data stored in the cache, that a second
more recent version of the data is available;
in response to detecting, independent of any requests for data stored in the cache,
that the second more recent version of the data is available, causing a first
older version of the data to be deleted from the cache, and requesting the
second more recent version of the data be supplied to the cache.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None